Survey of different Trajectory Planning Techniques of Mechanical Manipulators

Narender Kumar[1], Masood Ahmad[2]
[1,2]Department of Computer Science and Engineering, H.N.B. Garhwal University, Srinagar Garhwal, Uttarakhand, India
(E-mail: quest.masood@gmail.com)

Abstract— The trajectory is a history of position and its higher derivatives of time described using independent variables such as the degree of freedom. Path planning or trajectory generation is the key to unlock robotics, animations and as well as artificial intelligence to some extent. This paper discusses various trajectory generation techniques of a mechanical manipulator from the classical straight line with parabolic blends technique to genetic algorithms to find smooth and optimized trajectory. Optimization of execution time, total energy and jerk. Some of these techniques enhances manipulators properties such as hybrid and redundant manipulators using Cartesian and joint space schemes. This paper also compares various techniques on the bases of their qualitative features and computational expenses.

Keywords— *Manipulators; Trajectory Generation; Path Planning; Optimization*

I. Introduction

Trajectory planning or generation is the movement of manipulator's end-effector from the initial position to the final position using intermediate points without any collision within certain dynamic constraints. The user does not need to write down complicated input and output functions for every small task. In order to move manipulators from an initial orientation to a final orientation of the end effectors, the designer needs to write all those complicated functions for the user. We can call this as user's constraints. There are other types of constraints [8] (chapter 7) like joint space constraints, Cartesian space constraints, constraints due to obstacles, etc. Joint space constraints such as limitation of joint actuator sensors. Cartesian constraints, due to the limitation of manipulators workspace. The simplest approach in trajectory planning is the use of kinematics constraints to form a polynomial equation which describes the motion of a manipulator [8] (chapter 7). More sophisticated trajectory methods are being developed such dynamic algorithms, random sampling scheme, and genetic algorithm.

Trajectory planning does not have any basic principles. According to Alexander Stoytchev, the verification principle can be treated as a basic principle of developmental robotics [1]. It states that a system can produce and maintain only verified knowledge and the system should itself verify information that it collects. This principle is being used in trajectory generations as in the form of genetic algorithms [4] [11] and different real-time algorithms [6] [16]. Now, obvious questions can raise, "How to optimize a trajectory planning method?" Trajectory optimization is also an important part of trajectory generation, which deals with minimizations (or maximizations) of some performance within some constraints. Trajectory optimization is successfully being used in rocket science, industrial robots, and missile launching. It is well-proved fact that jerk limit is a necessity to prevent wear and tear of a manipulator and its end effector [14].

This introductory section tries to cover various qualitative problems that occur in trajectory planning to understand overall aspects. This paper presents the following sections. Section II contains basic working equations used in trajectory planning techniques, section III present literature review of a various paper on trajectory planning methods, section IV compares different techniques, section V concludes this paper and at last, but certainly not the least section VI is about future works in this area.

II. Working Equations

As user point of view, trajectory generator takes input as Cartesian position and give desired motion as an output, but if we look at path planner point of view Cartesian positions are converted into joint positions using inverse kinematics. Then using a technique like joint space trajectory planning desired joint sets are calculated kinematic set of the joint scheme. Trajectory controller uses this joint set and produces torque required for desired motion. Even we can also calculate desired orientation using Cartesian space scheme, but it has computationally expensive and adds to that path update rate is quite high in trajectory generation, hence this constraint puts Cartesian space scheme in a sideline.

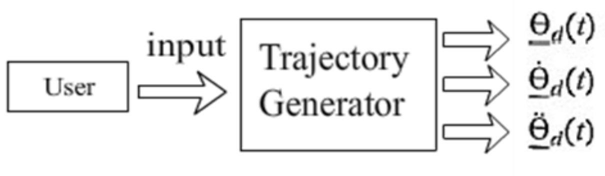

Figure 1: Basic Working of the Trajectory Generation

Some of the interpolation equations used in trajectory generations are –

Simple straight line spline function – It is one of the simplest possible equation, but its second derivative is not continuous at its edges (via points). This property is not the only reason for the low usage of this technique in trajectory planning. It neither provides constraints on velocity nor on acceleration.

E.g. $y = ax + b$.

Quartic straight line spline function – Second order equation is next in line. It provides velocity constraint but would not able to provide acceleration constraints.

E.g. $y = ax^2 + bx + c$.

Cubic straight line spline function – Cubic equations are most popular in trajectory generations, as these are linear in their second derivative and hence provide smooth acceleration.

E.g. $y = ax^3 + bx^2 + cx + d$.

These are most common interpolation functions in trajectory, their higher order is also possible but to solve them it required a high number of equations as it contains large number 'constraint coefficients'.

III. LITERATURE REVIEW

Table 1: Review of Different Trajectory Planning Techniques

S.No.	Paper Title	Review
I.	Planning and Execution of Straight Line Manipulator [13].	This paper discusses two techniques to achieve straight line motion, one deals with the real-time computations and other deal with precomputations. In a real-time approach, intermediate joint sets are calculated from the Cartesian sets at regular intervals using inverse kinematics. It offers various advantages over traditional techniques, including smooth planning, fewer computations. In the precomputation approach, motion planning proceeds on precomputed via points for motion generation which leads to the bounded deviation joint path. This method reduces real-time computations.
II.	Joint Trajectory of Mechanical Manipulators for Cartesian Path Approximation [3].	Methods discussed in this paper uses X – spline and quartic spline functions in trajectory generations. This is one of the 1980s paper which uses sequences of Cartesian points and using inverse kinematics convert them into the joint scheme. X – Spline is special interpolate cubic spline function which requires only one point for fitting the trajectory curve. Spline functions are calculated using interpolation of joint positions and velocities. Dynamic constraints are used in calculating spline functions. Trajectory points are calculated either using past or two future points (predefine via points). It provides a better result than polynomial trajectory generations. All via point's values are calculated even before calculating spline function.
III.	Optimal Trajectory Generation for Robotic Manipulators Using Dynamic Programming [2].	This paper formulates the idea of optimal trajectory planning as an optimal control technique with dynamic programming. This technique uses one moving manipulator to specify the trajectory. Manipulator move on a planned trajectory using feedback algorithm. It uses a recursive dynamic approach for fast and accurate convergence of the solutions. Computational efficiency is upgraded using dynamic constraints and recursive refinement procedure. This paper also provides the simulation of various effective methods.
IV.	Planning of Minimum-Time Trajectories for Robot Arms [7].	This paper began with flash back about trajectory optimization. Time optimization of a trajectory is always long unsolved standing problem, from the 1980s. Many researchers try to optimize the time execution algorithm, Lul and Walker in 1977 try to minimize the total time between two points using sequences of time intervals. Later Kim and Shin in 1985 uses Taylor's algorithm and introduces dynamics on accelerations constraints. This paper discusses a dynamic approach and a graph search algorithm to optimize the classic time scaling problem. The technique shows that optimal path nearly trends to a straight line in joint space scheme. Dynamics was either assume or calculated linearized to apply control theory. This paper also gives real-time approach or path planning. Due to heuristic nature, it is possible to get feasible solutions insisted on the optimal solution.

V.	A new Method for Smooth Trajectory Planning of Robot Manipulators [6].	This paper contains a smooth trajectory planning method of robot manipulators. Objective functions are created and contain terms directly proportional to the square of jerk and term proportional to total execution time to ensure smoothness of trajectory. Overall trajectory uses fifth – order B splines technique. It put an upper limitation of position, velocity, acceleration, and jerk. This technique works on minimization of objective function hence jerk and execution time. The algorithm uses kinematics constraints and objective functions as input and optimization are obtained using a Sequential Quadratic Programming technique.
VI.	Fast Smoothing of Manipulator Trajectories using Optimal Bounded-Acceleration Shortcuts [10].	As the name suggests this paper discusses a technique which tries to find a shortcut between two points on a trajectory. The method uses two via points, try to fit a straight line between them if possible replace intermediate trajectory with this new shortest possible trajectory or else reduce the distance between used via points and repeat this process. This method may not provide an optimal solution but provides one of the best feasible solutions with safety speed and aesthetics. This method also used in animations. Minimization technique uses the iterative numerical technique by replacing path of path shorter segment. It is also an online path generations technique. The main objective of this paper is to provide qualitative time optimal, bounded velocity and accelerations trajectory.
VII.	Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications [14].	Jerk may lead to wear and tear in a manipulator, its necessity to control it. This paper discusses a technique which uses fifth order polynomial to generate a smooth trajectory between two via points. Fifth order polynomial has six coefficients, hence twelve constraints are possible between two via points that are enough to present third derivatives of motion a jerk. The straight-line segment with parabolic bends combines with a quartic polynomial that constraint from oscillations and sine wave templates is used to calculate the end conditions from zero acceleration to non-zero acceleration and vice-versa. Reduces excitations of resonant frequencies, hence reduces actuator wears. This present another online trajectory generation technique but only for one degree of freedom. Limited jerk trajectories can be computed at speed and more accurately. As it is real-time computational, hence has low computational complexity. It also provides smooth joint to joint motion with jerk limits.
VIII.	Online Trajectory Generation: Straight-Line Trajectories [15].	This paper extends the idea of time-synchronized trajectory planning with phase – synchronized (popularly known as ‘homothetic trajectory’) to generates trajectory within low-level path update rate. This paper also provides a method that unable manipulators to react instantaneously to different real-world sensor events. Via points are calculated instantaneously and are used to generate next via points using feedback mechanism. Time-synchronization is achieved in three steps, calculation of synchronization time of each degree of freedom, synchronization of a selected degree of freedom and calculation of parameter of a via points. Homothetic (phase synchronization) required further three conditions selections of references of the degree of freedom, setting time synchronization to a minimum and updating constraints at each path updates.
IX.	Trajectory Planning of Redundant Manipulators using Genetic Algorithms [11].	Redundancy is the use of more degree of freedom than required for a given task. This paper extent genetic algorithms with pseudoinverse. A pseudoinverse is a generalized form of an inverse matrix, commonly used for a finding optimal solution to linear equations. Inverse kinematics calculate an infinite number of solutions and optimal need to be selected. Pseudoinverse guaranties generate end-effectors velocity with a minimum norm joint velocities and form a locally cyclic. This technique may lead to singularity reduction. This method able to deal with mechanical singularities but typical algorithm singularities may arise in constraint resultants of a mechanical manipulator.

X.	STOMP: Stochastic Trajectory Optimization for Motion Planning [12].	Stochastic is random probability distribution or able to locate statistically but can't precise. This technique key goal to create smooth motion with obstacle avoidance. Generate noisy explore in surrounding space and update trajectory with lower cost. The cost function is optimized using iterations. This technique also overcomes local minima. This technique does not require the cost function gradient and works on high dimensions robotics. So, this technique overcomes the limitation of local gradient technique used in CHOMPS. Path update rate under some assumptions guaranties non-increasing cost function. This technique is based on reinforcement learning, which requires real-time computations and executions.
XI.	Manipulator trajectory planning using a MOEA [4].	Moving towards multiple objective and multiple avoidances. This paper address the genetic algorithm for trajectory planning. The technique effectively works in two and three degrees of freedom and up to five simultaneously objectives. This paper also present trajectory optimization of 2R and 3R robotic manipulators with 2D (objectives), 2R manipulators with 5D (objectives) and 3R manipulator with 5D (objectives). It also compares in between different objectives, e.g. clockwise movement in 3R manipulator (5D) differ by joint distance as compare to anti-clockwise movement.
XII.	A Random Sampling Scheme for Path Planning [9].	This paper presents a generalized planning scheme that extends the idea of a random sampling of manipulator's configuration space using probabilistic complete. Technique discusses in this paper is collision-free path planning under the assumptions of known surrounding knowledge. The module of the basic problem is upgraded to computational complexity. Probabilistic complete means if there is a possible path exists between two via points then find one with the highest probability. Probabilistic completeness is achieved either by potential field planner or by precomputing a roadmap. This paper also discusses CAD interface and applying it with animations for trajectory planning.
XIII.	Collision free trajectory planning for hybrid manipulators [5].	This paper present optimal trajectory planning in the presence of obstacles using a quick random search algorithm with optimization by genetic algorithm. This paper also discusses the hybrid of two different architecture, serial and parallel architecture. The objective function contains a genetic algorithm and random search properties. i. Usually, roots are made of serial architecture. a. Low accuracy and stiffness b. Low load/ mass ratio and heavy mass ii. Parallel architecture a. Low mass compact structure, better accuracy, and stiffness. b. Complex design architecture c. Difficult kinematics and presence of singularities The technique takes advantage of both techniques, rapidly-exploring random trees, and optimization of the genetic algorithm for optimizing planning.

IV. COMPARISON OF RESULTS

This section contains a qualitative comparison of different trajectory techniques discussed in this paper. The qualitative comparison is based on the following aspects –

i. Optimization

ii. Constraints

iii. Trajectory Functions

Optimization is a primary requirement of trajectory planning. Execution time is one of the criteria for optimal solutions [2] [4] [7] [15]. Another criterion can be space optimization, due to real-time computing and a low number of constraint space optimization has not been the topic of discussion for the time, but increasing constraints and real-time computations will eventually lead to the space optimization problem.

Constraints are ingredients of trajectory, a recipe is made up of ingredients similarly trajectory is made of constraints. One can change the taste of a recipe by changing amount of ingredient as one can change trajectory by changing constraints. Constraints can further be divided into dynamic constraints and hardware constraints. Dynamic constraints are used to define properties of position, velocity, acceleration and their higher derivative such as initial velocity and final velocity will be zero or velocity at some points is fixed. Hardware constraints are a limitation due to hardware used such as the maximum velocity of an actuator.

Trajectory function is a curve that defines between two via points it can be a straight line [13] [16], a quartic [3], or a higher order function [4] [6] [12] [14] [15]. The objective function is key to achieve optimization.

Papers [2] [3] [13] define basic those are widely used in the mechanical manipulator. Paper [13] define straight line planning, [3] uses a joint scheme and [2] recursive dynamic programming for planning trajectory. Papers [6] [7] [10] [14] [15] discusses optimization. Paper [7] try to find a time-optimal trajectory, [6], and [10] uses simple logic for defining smooth trajectory. Paper [14] put constraints on jerk but this technique is applicable to only one degree of freedom and [2] [7] [6] [10] [14] [15] uses real-time trajectory generations.

Genetic algorithm [4] [11] [12] have rapidly gained popularity due to close real-world connection hence better explain real-world problems and solutions. Genetic algorithms can easily deal with singularity problems using direct kinematics. Paper [9] explain the technique of random samplings. Collision free trajectory using a genetic algorithm is discussed in the paper [5].

V. FUTURE SCOPE

Further studies might consider using the different advance genetic algorithm in trajectory planning, dealing with uncertainty, like moving obstacle avoidance, reducing the complexity of trajectory algorithms. Applying motion trajectory planning to animations, autonomous aerial robots, advanced cloud computing, etc. are an area for future work.

REFERENCES

[1] Alexander Stoytchev, "Some Basic Principles of Developmental Robotics", IEEE Transactions on Autonomous Mental Development, Vol. 1, No. 2, August 2009.

[2] Atef A. Ata, "Optimal Trajectory Planning of manipulators", Journal of Engineering Science and Technology Vol. 2, No. 1 (2007) 32-54, School of Engineering, Taylor's University College, 2007.

[3] C.S. Liii, P.R. Chang, "Joint Trajectory of Mechanical Manipulators for Cartesian Path Approximation", IEEE Transactions on Systems, Man, and cybernetics, Vol. SMC-13, 1983.

[4] E.J. Solteiro Pires, P.B. de Moura Oliveira and J.A. Tenreiro Machado, "Manipulator trajectory planning using a MOEA", ScienceDirect, Elsevier, Applied Soft Computing, 7, (2007) 659–667, 2007.

[5] F. Gómez-Bravo, G. Carbone, J.C. Fortes, "Collision free trajectory planning for hybrid manipulators", Elsevier, Mechatronics 22 (2012) 836–851, 2012.

[6] Gasparetto, V. Zanotto, "A new method for smooth trajectory planning of robot manipulators", Mechanism and Machine Theory 42 (2007) 455–471, 2007.

[7] Gideon Sahar, John M. Hollerbach, "Planning of Minimum-Time Trajectories for Robot Arms", The International Journal of Robotics Research 1986 5: 90, DOI: 10.1177/027836498600500305, 1986.

[8] John J. Craig, "Introduction to Robotics Mechanics and Control (3rd Edition)", Pearson Education, Inc. (2005).

[9] J. Barraquand, L. Kavraki, J.C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan, "A Random Sampling Scheme for Path Planning", International Journal of Robotics Research, 16(6): 759—774, 1997.

[10] Kris Hauser, Victor Ng-Thow-Hing, "Fast Smoothing of Manipulator Trajectories using Optimal Bounded-Acceleration Shortcuts", IEEE International Conference on Robotics and Automation Anchorage Convention District, May 3-8, Anchorage, Alaska, USA, 2010.

[11] Maria da Graça Marcos, J.A. Tenreiro Machado, T.-P. Azevedo-Perdicoúlis, "Trajectory planning of redundant manipulators using genetic algorithms", Elsevier, Commun Nonlinear Sci Numer Simulat 14 (2009) 2858–2869, 2009.

[12] Mrinal Kalakrishnam, Sachin Chitta, Evangelous Theodorou, Peter Pastor and Stefan Schaal, "STOMP: Stochastic Trajectory Optimization for Motion Planning", IEEE International Conference on Robotics and Automation, Shanghai International Center, May 9-13, Shanghai, China, 2011.

[13] R. Taylor, "Planning and Execution of Straight Line Manipulator Trajectories", in Robot Motion, Brady et al., Editors, MIT Press, Cambridge, MA, 1983.

[14] Sonja Macfarlane, Elizabeth A. Croft, "Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications", IEEE Transactions on Robotics and Automation, Vol. 19, No. 1, February 2003.

[15] S. Singh and M. C. Leu, "Optimal Trajectory Generation for Robotic Manipulators Using Dynamic Programming", Journal of Dynamic Systems, Measurement, and Control, JUNE, Vol. 109, 1987.

[16] Torsten Kr¨oger, "Online Trajectory Generation: Straight-Line Trajectories", IEEE Transactions on Robotics, Vol. 27, No. 5, October 2011.